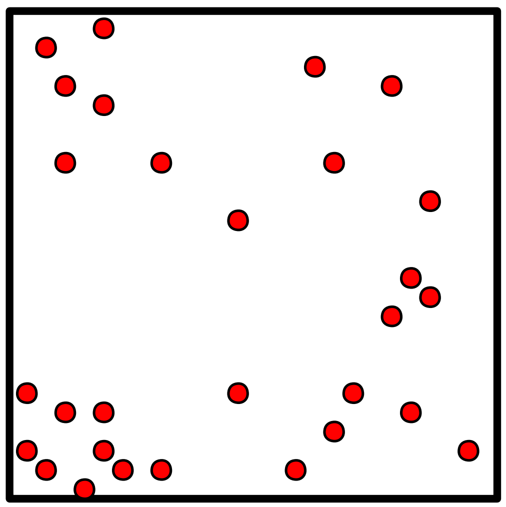
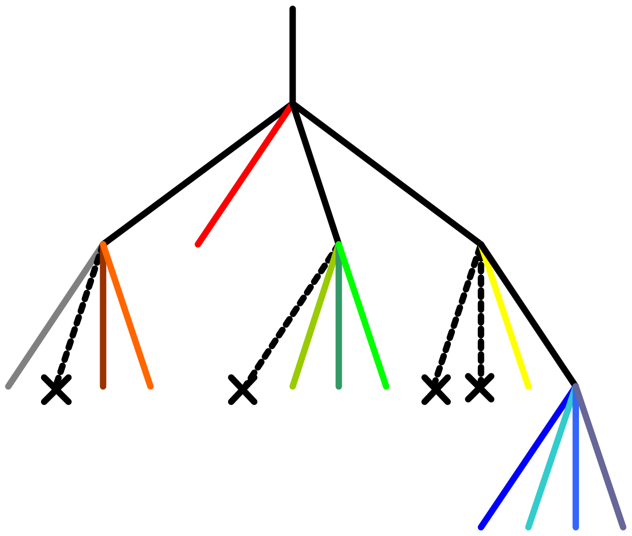
Recursive Bisection

Objective

Recursive bisection is commonly used to partition grids so that the workload is evenly distributed. It also has the advantage that locality in space corresponds to locality in memory, which aids performance on cache-based architectures.

The objective for this data set is to divide the particles into groups of four or less. The particles can represent anything from stars to atoms.





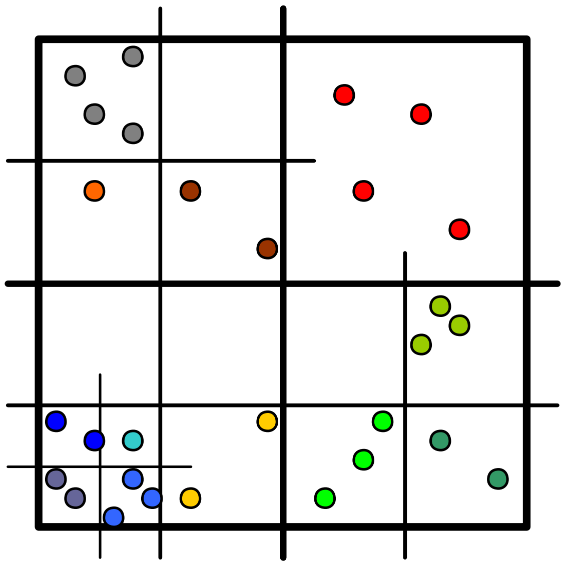
Method

In this 2D example, the parent process creates four threads, one for each quadrant. Each thread counts the number of particles (N) in its quadrant and performs one of three actions:

1. Exits, N=0
2. Begins computation, N≤4
3. Creates four new threads, N>4.

Result

After the last round of thread creation there are 12 threads performing the calculation. Their workloads are approximately balanced.



Quiz #5

1. For a 1D grid, how many threads should be created in each recursion step? How many for a 3D grid?
2. Can you think of a potential pitfall of this method? [Hint: Imagine a grid with millions of particles.]